IMPROVED CAPACITY REVERSIBLE WATERMARKING

D. Coltuc

Valahia University Targoviste, Romania
coltuc@valahia.ro

ABSTRACT

This paper continues our researches on high capacity reversible watermarking based on simple transforms. Image pixels obeying some simple constraints are transformed and then, data is embedded by simple additions. The transform induces a congruence equation. At detection, the transformed/not transformed pixels are found by checking the congruence, watermark codes are extracted and original image is recovered. The proposed method is of very low mathematical complexity and does not need additional data compression. The major novelty of the proposed technique is the use of a different transform which allows the embedding of watermark codewords into each transformed pixel. Thus, the improved version provides considerably higher embedding bit-rate. Experimental results for a single embedding level are presented.

Index Terms— reversible watermarking, high capacity

1. INTRODUCTION

We have recently proposed a high capacity low cost reversible watermarking scheme based on a simple integer transform defined on pairs of pixels [1]. The transform introduced in [1] generalizes the one of [2, 3] and introduces a different principle for marking/detection based on divisibility. A parameter n is introduced providing, for a single embedding level, a theoretical maximum data hiding capacity of $\log_2(2n)/2$ bits per pixel. For $n > 2$, bit-rates greater than 1 bpp can be obtained in a single iteration and without any additional data compression. Besides, the data embedding procedure based on integer divisibility eliminates the need of storing a map of transformed pairs. The spatial domain reversible watermarking based on such simple integer transforms provides almost similar results as the difference expansion ones [4, 5], but at a considerably lower mathematical complexity. We should mention that with respect to the embedding bit-rate, the difference expansion watermarking provides slightly higher bit-rates.

This paper continues our researches on reversible watermarking. The scheme proposed in [1] is revisited in the context of a different transform. While for the previous scheme, a single watermark codeword is inserted into a pair of transformed pixels, the new one allows codeword embedding into each transformed pixel. For a single level of embedding, the theoretical maximum capacity of the corresponding watermarking scheme is of $\log_2 n$ bpp. The experimental results obtained so far are very promising: the new scheme appears to outperform the high capacity reversible schemes reported so far in the literature. The interesting features of the reversible contrast mapping watermarking are preserved, namely very low mathematical complexity, spatial domain marking, multilevel marking.

The outline of the paper is as follows. The new integer transform and the watermarking scheme are introduced in Section 2. The improvements of the new scheme with respect to the previous one are discussed in Section 3. Experimental results are shown in Section 4. Finally, the conclusions are drawn in Section 5.

2. REVERSIBLE WATERMARKING SCHEME

Let x_i, $i = 0, \ldots, N$, be image pixels. Let $[0, L]$ be image graylevel range ($L = 255$ for 8 bit graylevel images) and let $n \geq 1$ be a fixed integer. Let us further consider the following transform defined on the sequence of pixels:

$$y_i = (n+1)x_i - nx_{i+1} \tag{1}$$

Let us impose the constraint to transform pixels into pixels, i.e., a pixel x_i is transformed if and only if:

$$0 \leq (n+1)x_i - nx_{i+1} \leq L \tag{2}$$

Therefore T is defined on a restriction of the Cartesian product $[0, L] \times [0, L]$

The transform is invertible. By simple calculus, one has its inverse:

$$x_i = \frac{y_i + nx_{i+1}}{n+1} \tag{3}$$

Equation (3) exactly inverts (1). Since x_i, y_i, x_{i+1} and n are integers, the above equation shows that $n+1$ divides $y_i + nx_{i+1}$. The divisibility can be written as a simple congruence equation:

$$y_i + nx_{i+1} \equiv 0 \bmod (n+1) \tag{4}$$

Stated in other words, if a pixel x_i has been transformed by equation (1), the transformed pixel y_i obeys equation (4).

As in [1], the divisibility is the essential property for reversible data hiding. The transformed pixels obey (4), while the not transformed ones does not necessarily satisfy (4). Before going any further, let us observe that the transformed pixels can be simply modified in order to not fulfill equation (4). Meantime, the not transformed pixels can be modified in order to satisfy the divisibility equation.

Let us first consider a transformed pixel and let us modify it by simple adding an integer in $[1, n]$. Let w_i be such an integer and let us add w_i to y_i:

$$y_i \rightarrow w_i + y_i \tag{5}$$

By introducing (5) into (4) one has: $w_i + y_i + nx_{i+1} \bmod (n+1) = w_i$ Since $w_i \in [1, n]$, the divisibility fails . To prevent pixel overflow, a supplementary constraint must be imposed:

$$y_i + n = (n+1)x_i - nx_{i+1} + n \leq L \tag{6}$$

Let us next consider a not transformed pixel x_i and let c_i be as follows:

$$c_i = (x_i + nx_{i+1}) \bmod (n+1) \tag{7}$$

Obviously $c_i \in [0, n]$. Let us further simply subtract c_i from x_i:

$$x_i \rightarrow x_i - c_i \tag{8}$$

Since the pixel is not transformed $y_i = x_i$ and, from (4) one has $x_i - c_i + nx_{i+1} \equiv 0 \bmod (n+1)$, i.e., (4) is fulfilled. A similar effect is obtained by adding $(n+1-c_i) \bmod (n+1)$ to x_i, with $x_i + n \leq L$.

With the above considerations, the basic principle of the watermarking follows. First, the not transformed pixel pairs are modified in order to fulfill (4). Then watermark codewords are embedded into the transformed pixel pairs by using equation (5). Since the pixels with embedded watermark codewords no longer obey (4), they are immediately identified at detection. Furthermore, if the watermark contains the correction data as well, i.e., the c_i sequence, the original image can be completely restored with no loss of information. Obviously, the watermarking is possible if there are more transformed pixels than not transformed ones.

2.1. Marking

The marking procedure consists in 3 steps:

1. Define an one-dimensional indexing of the image: x_i, $i = 0, \ldots, N$;
2. For $i = 0$ to $N - 1$, check equations (2,6):
   (a) if equations (2,6) are fulfilled, transform x_i;
   (b) if equations (2,6) are not fulfilled, modify x_i as described by equations (7,8) and collect corresponding correction code c_i;
3. Create, encode and embed watermark into the transformed pixels starting from $i = N - 1$, in decreasing order of the index i.

In step 1, the one-dimensional indexing of image pixels can be obtained by scanning the image on rows, columns or on any other space filling curve. Theoretically any indexing can be used, but the one obtained by a space filling curve provides more data hiding capacity since it exploits the correlation between adjacent pixels.

In step 2, all the pixels are constrained to satisfy divisibility equation (4) either by transforming with equation (1), or by correcting with (7,8). In this step, a map containing the location of transformed pixels is created as well. The map is further used in step 3.

In step 3, the payload together with the correction data collected in step 2 should be embedded into the transformed pixels by equation (5). The watermark codewords should be integers in $[1, n]$. Therefore, the collected sequence should be first encoded to eliminate all the values $c_j = 0$. Since there are n allowed integer values to encode $n + 1$ values, one can simply use $n - 1$ integers to encode $n - 1$ values and the remaining available integer is used together with two other integers in $[1, n]$ to encode the remaining 2 values. The watermark embedding is performed in reversed order of the index since this is the ordering demanded at detection. The watermark, after some header codewords, contains the sequence of collected data c_i, in reversed order of their index, too. This organization of the data for the watermark is necessary to make available the correction data as required by the watermark detection stage.

2.2. Watermark detection and original recovery

The watermark is sequentially extracted and, simultaneously, the original image is recovered as follows:

1. Define the same one-dimensional indexing of the image as the one of the marking stage;
2. For $i = N$ to 0, check equation (4)
   (a) if equation (4) is not fulfilled, extract watermark codeword (correction code or payload code), subtract codeword from the current pixel and recover original pixel value by inverse transform (3);
   (b) if equation (4) is fulfilled, recover original pixel by adding the corresponding decoded correction code c_i;
3. Extract entire payload.

Let x_i be the current pixel. The original value of the pixel located at index $i + 1$ is already available (x_N is not altered by the marking algorithm). If the congruence fails, i.e., $x_i + nx_{i+1} \bmod (n+1) = w_i$ and $w_i \neq 0$, it follows that w_i is exactly the inserted codeword into the transformed pixel located at index i. By subtracting w_i from the current pixel one obtains the transformed pixel y_i. Then, the original pixel is obtained from equation (3). The extracted watermark codeword is either a correction code c_i or, after correction codes are exhausted, a payload codeword.

If $x_i + nx_{i+1} \mod (n+1) = 0$, the current pixel was not transformed. It was modified to fulfill (4). The corresponding correction codeword should be extracted from the collected watermark sequence, decoded and then, simply added to the current pixel to recover the original image pixel.

2.3. Hiding Capacity

Let t be the number of transformed pixels. Each transformed pixel allows the insertion of a codeword in the range $[1, n]$, namely the insertion of $\log_2(n)$ bits. Each one of the $N-t$ not transformed pixels should be corrected by an integer ranging in $[0, n]$, i.e., by $\log_2(n+1)$ bits. The theoretical bit-rate of the proposed reversible watermarking scheme, $b(n)$, is:

$$b(n) = \frac{t}{N}\log_2(n) - \frac{N-t}{N}\log_2(n+1) \text{ bpp} \qquad (9)$$

When the number of transformed pixels is high, namely t is approximately N, one has the upper bound of the proposed scheme:

$$b(n) \leq \log_2(n) \text{ bpp} \qquad (10)$$

Equation (10) shows that if the number of transformed pixel pairs is large enough, the proposed scheme can provide more than 1 bit per pixel bit-rate, as soon as $n > 2$. For $n > 4$, the bit-rate can be greater than 2 bpp, provided that the number of transformed pixels is large enough.

3. A COMPARATIVE STUDY

The novel watermarking scheme follows the same basic principles as the one proposed in [1]: transform pixels to satisfy a divisibility equation, modify the pixels which cannot be transformed and embed data into the transformed ones. The transform used in [1] is:

$$\begin{aligned} y_i &= (n+1)x_i - nx_{i+1} \\ y_{i+1} &= -nx_i + (n+1)x_{i+1} \end{aligned} \qquad (11)$$

While equation (11) transforms simultaneously both x_i and x_{i+1}, the new one transfoms only x_i. This is the major novelty that allows a significant gain in embedding capacity. Thus, after x_i is transformed, the new scheme transforms x_{i+1} and so on. Furthermore a codeword of $\log_2 n$ bits is embedded into each transformed pixel, while equation (11) embeds a single codeword of $\log_2 2n$ bits for a pair of pixels. Since $2\log_2 n > \log_2 2n$ as soon as $n > 2$, the novel scheme is expected to outperform [1].

Since the new watermarking scheme allows the embedding of integer codewords ranging in $[1, n]$, it provides data hiding capacity only for $n > 1$. The previous scheme provides data embedding capacity even for $n = 1$, case corresponding to watermark codewords $\{1, 2\}$.

For both transforms, the error between y_i and x_i is the same: $n(x_i - x_{i+1})$. Therefore, the gain in bit-rate is expected to be obtain at rather similar distortion.

The new scheme transforms only a pixel instead of a pair. This yields to slightly simpler correction code computation as well as watermark codeword extraction. There are some other differences for both marking and detection procedures. For instance, the newly proposed scheme performs simultaneous watermark extraction and original recovery, while the previous one first performs watermark extraction and then, original recovery. The mathematical complexity of both schemes is very similar. As already stressed, the newly proposed scheme preserves the lower computational complexity of the previous one and achieves high embedding bit-rates without any additional data compression.

4. EXPERIMENTAL RESULTS

The embedding capacity of the proposed scheme depends on the number of transformed pixels (i.e., on image statistics) and on the parameter n. We investigate the dependence on n of the reversible watermarking scheme and we compare the results with the ones of the previously reported version. As in [1], we consider only the first level of embedding (i.e., a single pass of the algorithm).

Let us consider the test image *Lena*. A single pass of the proposed algorithm for $n = 2$ gives a bit-rate of 0.96 bpp. The bit-rate is almost the same of [1], but at a lower PSNR (22.85 dB compared with 25.24 dB). For $n = 3$ one gets 1.46 bpp at 20.15 dB which already equals the maximum bit-rate obtained with the scheme of [1], namely 1.42 bpp at 19.95 dB (obtained for $n = 6$).

By increasing n, the bit-rate increases: for $n = 4$ one gets 1.77 bpp, for $n = 5$ the bit-rate is 1.97 bpp, for $n = 6$ the bit-rate is 2,08 bpp and so on, up to the maximum value of 2.19 bpp obtained for $n = 9$. The further increase of n does not provide any improvement: the decrease of the number of transformed pixels is more significant than the increase of the number of bits of the corresponding codewords.

Details of the original test image *Lena* and of some watermarked versions are shown in Fig. 1. The bit-rates for a single pass of the algorithm for n ranging in [2,24] are plotted in Fig. 2: the stars represent the new scheme results, while the diamonds the already reported version ones. As it can be seen, the improvement in data hiding capacity is significant. Compared with the theoretical upper bound of $\log_2 n$, the results are close for small values of n ($n = 2$, $n = 3$). As n increases, $N - t$ becomes non negligible in (9) and the bit-rate decreases with respect to the maximum bound.

The number of transformed pixel pairs, t, depends on image statistics. For the highly textured image *Mandrill*, the bit-

(a) Original

(b) $n = 3$, $b(3) = 1.46$ bpp

(c) $n = 4$, $b(4) = 1.77$ bpp

(d) $n = 7$, $b(7) = 2.16$ bpp

Fig. 1. Details of original (a) and marked (b-d) versions.

rates are considerably lower: 0.75 bpp ($n = 2$), 1 bpp ($n = 3$, $n = 4$). Furthermore, as n increases, the bit-rate decreases: 0.88 bpp for $n = 5$ and so on. The results are considerably improved compared with [1], where the maximum bit-rate is only of 0.5 bpp. On the contrary, for images having large uniform areas, one can expect, in a single pass, bit-rates much higher than 2 bpp.

5. CONCLUSIONS

The major novelty of the proposed technique is the use of a slightly different transform which allows data embedding of watermark codewords into each transformed pixel. Compared with our previous schemes where a single codeword is embedded into a transformed pair of pixels, the gain in data hiding capacity is significant.

The results obtained so far are very promising. Thus, for the test image *Lena*, compared with [1], the improvement is of 0.77 bpp. For the same test image, the results obtained in a single pass of our scheme appear to outperform the results provided by the high-capacity schemes reported so far in the literature. The novel scheme outperforms with 0.33 bpp the results reported in [3] and with 0.22 bpp the results obtained by difference expansion [4]. It should be mentioned that we compare the results obtained only for the first level of embedding of the new algorithm, while in [3, 4] the reported results are the maximum bit-rates obtained after multiple levels of embedding.

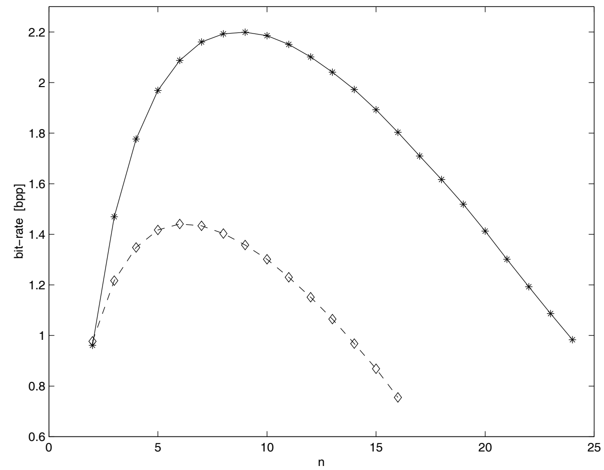

Fig. 2. Bit-rate with respect to n for *Lena* in a single pass of the scheme: proposed scheme (*), previous version ($\diamond$).

The proposed scheme works into the spatial domain and it is of very low mathematical complexity. No additional data compression is necessary. In this paper, only the bit-rate obtained for a single pass of the algorithm is investigated. A second pass of the algorithm brings some more hiding capacity. For instance, for the test image *Lena*, the bit-rate after two reversible embedding stages is greater than 2.5 bpp.

Acknowledgment

This work is supported by INFOSOC (CEEX 2006) Romania, in the frame of the project 3DRoboVis.

6. REFERENCES

[1] D. Coltuc, J-M. Chassery, ”High Capacity Reversible Watermarking”, *Proceedings of the IEEE International Conference on Image Processing ICIP’2006*, Atlanta, GA SUA, pp. 2565-2568, 2006.

[2] D. Coltuc, A. Tremeau, ”Simple Reversible Watermarking Schemes”, *SPIE Proceedings*, vol. 5681, pp. 561–568, 2005.

[3] D. Coltuc, J-M. Chassery,, ”Very Fast Watermarking by Reversible Contrast Mapping”, *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.

[4] J. Tian, ”Reversible Data Embedding Using a Difference Expansion”, *IEEE Trans. on Circuits and Systems for Videotechnology*, vol. 13, no. 8, pp. 890–896, 2003.

[5] A. M. Alattar, Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform, *IEEE Trans. on Image Processing*, vol. 13, no.8, pp. 1147–1156, 2004.